N.15 Sped in abb. post. Gr. III L. 9.000

10 PROGRAMMI SU CASSETTA

BIOS UTILITY
WP: OPERAZIONI SUL TESTO

GZOOOOZ

ASCII SOFTWARE COURTESY

MISTER KIT

Elettronica 2000

ELETTRONICA APPLICATA, SCIENZA E TECNICA

N. 94 ... 1987 - L. 3.500
... post. gruppo III

OGNI MESE IN EDICOLA!

fantastico
LASER
LIGHTS
& MEDICAL

METAL DETECTOR

TESTER QUARZI

DIGITAL VOLTMETRI

TOUCH CONTROL

RIVELATORE INFRAROSSI

MODEM NEWS

MSX Computer Magazine è edita da Arcadia srl,
C.so Vitt. Emanuele 15, Milano.
Tel. 02/706329 (solo giovedì h. 15-18).
Una copia L. 9.000.
Fotocomposizione: Composit.
Stampa: Garzanti,
Milano. Distribuzione: SO.DI.P. Angelo
Patuzzi srl, Via Zuretti 25, Milano.
Registrato Trib. Milano N. 52 del 2/2/85.
Resp. Sira Rocchi.
Sped. in abb. post. Gr. III/70.

IN QUESTO NUMERO

★ WP: LE OPERAZIONI SUL TESTO

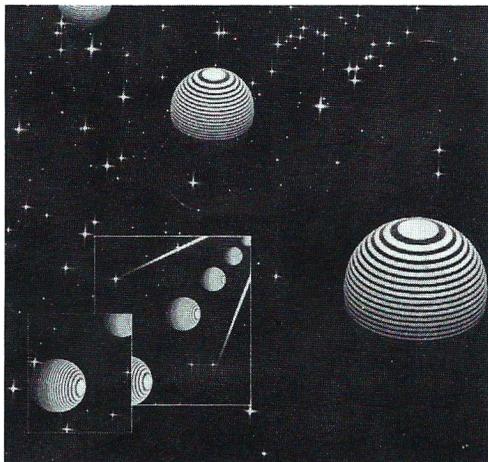
★ BIOS UTILITY

★ 10 PROGRAMMI 10

- BANANA
- CAR RACE
- POKER CASINÒ
- MSX VOICE
- ASSEMBLER ESPANSO
- GOLF 3D
- GOAL
- DOUBLE ROTATION
- CALCULATOR
- CORSO DI DATTILO

MSX TAPE SOFT

BANANA
(BLOAD "CAS:BANANA",R - 32 K)

I programmi contenuti in questo fascicolo di MSX COMPUTER MAGAZINE sono tutti compatibili con qualsiasi sistema MSX. Ecco per voi ancora 10 bei programmi! Ricordate di collegare la spina del controllo motore alla presa REM del vostro registratore se quest'ultimo la possiede. Assicuratevi che la spina nera sia collegata alla presa EAR del registratore e che la spina rossa sia inserita nella presa MIC. Se il vostro mangiacassette non possiede la presa REM, fate particolare attenzione a quando un programma è stato caricato o deve esserlo, affinché il nastro scorra per il giusto tempo. Appena vedete apparire sul video, dopo un comando di caricamento, la scritta OK, spegnete il registratore.

Nella cassetta allegata a questo fascicolo troverete, sul lato A:
Banana, Car Race, Poker Casinò, MSX Voice, Assembler espanso.
Sul lato B ci sono:
Golf 3D, Goal, Double Rotation, Calculator, Corso di dattilo.

Continua la storia di Indiana Jones con questo divertentissimo gioco nel quale rivestite i panni di un esploratore che, allontanatosi dalla spedizione, si trova inseguito da un discreto numero di affamatissimi cannibali. Starà a voi giocare ad una specie di rimpiattino saltellando di sasso in sasso su di un fiume, inseguiti da signori poco vestiti ma molto affamati. Vostre uniche armi di difesa le banane, che potete lasciar cadere (premendo il tasto «X») sul sasso sul quale vi trovate. Scopo del gioco è infatti quello di far cadere i cannibali in acqua e procurare che la corrente del fiume li trascini via.

Un altro modo per provocare la caduta dei cannibali è quello di raggiungere la riva destra, prelevare i sassi che vi si trovano e tirarli addosso ai

nemici. Questa operazione deve essere fatta nello stesso modo che usate per spostarvi, cioè utilizzando i tasti cursore. Più precisamente, lo spostamento del vostro personaggio avviene puntando la freccia sulla sommità del sasso sul quale volete saltare e premendo successivamente il tasto «C».

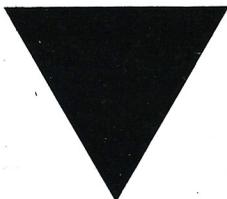
Quando i sassi sulla riva finiscono bisogna tornare sul fiume e poi dirigersi nuovamente verso la riva.

Attenzione: per caricare il programma usate il comando BLOAD "CAS:BANANA", R.

Buon divertimento e... in bocca al cannibale!

CAR RACE

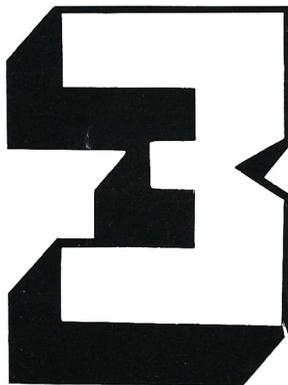
(CLOAD "CAR" - 16 K)

adatt. da E. Dassi

Il traffico stradale aumenta sempre di più e, di conseguenza, è sempre più difficile e pericoloso spingere la propria vettura a velocità sostenuta, a meno che non si corra in pista o... si scenda dall'automobile e ci si metta a giocare a questo video-game!

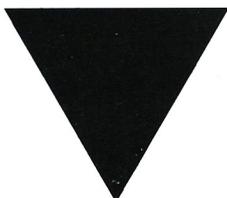
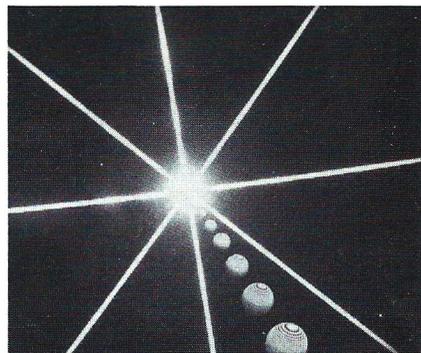
Scopo del gioco è riuscire a terminare il percorso di gara nel minor tempo possibile.

Appena caricato il programma dare il RUN, attendere qualche istante e, quando appare la scritta «CAR RACE», premere la barra spaziatrice (o il tasto fire se si usa il joystick). Inserite quindi il livello di gioco (da 1 a 4), che corrisponde ad una delle quattro piste.

A questo punto sul video appariranno il tracciato di gara ed un'automobilina da corsa tipo quelle da Formula 1. Per dosare il gas della vettura premere il tasto spazio (o il tasto fire del joystick); per cambiare la direzione di marcia usare invece i tasti cursore destra e sinistra. La gara si conclude quando vengono percorsi 5 giri della pista.

L'intera grafica del gioco è realizzata in screen 1. I percorsi di gara sono stati costruiti componendo insieme alcuni caratteri (vedi linee 1800-2730), quindi non è difficile inserire nel programma altri tracciati di gara.

POKER CASINÒ

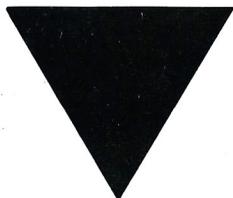
(CLOAD "POKER" - 16 K)

di P. Tamburelli

Vi trovate al Casinò e volete provare a giocare d'azzardo. La cosa migliore è tentare con il poker elettronico, il cui funzionamento è molto semplice: quando vi sarà richiesta la puntata dovrete battere la somma che intendete giocare e, poiché l'importo

dovrà essere sempre di tre cifre, nel caso vogliate giocare meno di 100 $ dovrete battere lo 0 come prima cifra (se volete giocare 20 $, per esempio, dovrete scrivere 020). Il calcolatore vi darà dunque le carte e vi chiederà quante e quali volete cambiare: per rispondere basterà semplicemente premere il numero corrispondente alla carta o alle carte che si vogliono sostituire. Dopo aver fatto questa operazione, dovrete premere lo 0 per consentire al calcolatore di riordinare le carte secondo la nuova combinazione e, se sarete fortunati, di pagarvi la

vincita. Per calcolare l'importo in caso di vincita, il calcolatore si basa sui valori descritti nella tabella in alto a destra dello schermo.

La strutturazione del programma è piuttosto semplice. La creazione delle carte è divisa in semplici print per i simboli grafici e in alcune centinaia di comandi vpoke, posizionate da opportuni calcoli, per le figure; il tutto corredato da alcune istruzioni line,bf le quali colorano più parti dello schermo.

Buona fortuna!

MSX VOICE
(CLOAD "V-PROC" - 32 K)
di G. Barfi

Questo programma si occupa del campionamento di suoni provenienti dal registratore a cassette, quindi è consigliabile predisporre un nastro sul quale ci siano già i suoni che vogliamo digitalizzare. Con MSX Voice si possono inserire in memoria una parte di un brano musicale, oppu-

re la propria voce. Ciò è possibile perché qualsiasi suono può essere tradotto in una sequenza di bit. Il metodo usato dal programma per digitalizzare i suoni consiste nel leggere un impulso dal registratore (PSG — registro 14 — bit n. 7) e nel salvarlo in memoria. Queste operazioni vengono ripetute molto velocemente (circa 28000 letture al secondo). Naturalmente più si aumenta la fedeltà di riproduzione, maggiore è la quantità di memoria utilizzata.

Il programma si avvale di tre routine in linguaggio macchina riguardanti l'input-output del suono e di una routine, sempre in linguaggio macchina, che si occupa dello scrolling pixel per pixel in screen 2.

Per caricare il programma, digitare CLOAD «V-PROC». Appena appare la scritta «OK», digitare RUN ed attendere il caricamento automatico dei dati registrati con l'istruzione BSAVE. Per accedere al menu premere il tasto funzione F1.

Vediamo le 7 opzioni disponibili:

1) Input-Output: questa opzione consente di digitalizzare in tempo reale i suoni provenienti dal registratore. Il computer invierà direttamente i suoni digitalizzati all'uscita audio, alla quale avremo collegato l'altoparlante del televisore o del monitor. Nessun dato verrà conservato in memoria.

2) Input-Memory: serve a prelevare i suoni dal registratore ed a memorizzarli in memoria, più precisamente nell'area RAM definita con l'opzione «Def Memory». Prima di digitalizzare l'input, viene richiesta la velocità di elaborazione. Minore è il valore immesso, superiore sarà la qualità di digitalizzazione e la quantità di memoria usata.

3) Memory-Output: va eseguita dopo l'opzione n. 2 (oppure dopo l'opzione n. 6) perché consente di riprodurre i suoni memorizzati in RAM. Prima di ascoltare l'uscita sonora digitalizzata, bisognerà immettere la velocità di output la quale dovrà essere, se si vuole una riproduzione normale, pari a quella di input.

4) Def Memory: con questa opzione viene definita la dimensione dell'area di memoria da utilizzare per la memorizzazione dei dati. È ovvio che, più quest'area è estesa, maggiore è il tempo di campionamento. Per chi ha collegato il disk drive, il limite massimo di memoria non è l'indirizzo 62335 ma il 57975.

5) Save Voice: serve per salvare su qualsiasi supporto magnetico (registratore, QuickDisk o Floppy) l'area di memoria stabilita nell'opzione «Def Memory». Il nome assegnato al file dei dati è «DATI». Per ogni blocco di memoria salvato è consigliabile annotare gli indirizzi di partenza e fine, nonché la velocità di esecuzione.

6) Load Voice: questa opzione carica in memoria il file dei dati digitalizzati, registrati con l'opzione n.5.

7) Fine: ritorno al Basic.

ASSEMBLER ESPANSO
(CLOAD "ASSEME" - 64 K)
di D. Montresor

Attenzione: per caricare il programma bisogna digitare CLOAD "ASSEME". Una volta apparsa la scritta 'OK', dare il RUN ed attendere che vengano caricate le due parti di programma che seguono. Funzione di questa utility è caricare in memoria dei programmi scritti in Assembler Z80 ed eseguire su di essi, oltre che l'assemblaggio, numerose operazioni.

Quando appare la prima riga vuota si può cominciare a scrivere i mnemonici oppure passare al menu premendo il tasto SELECT. I valori o indirizzi esadecimali necessari per alcune istruzioni devono sempre avere: due cifre per valori ad un byte e quattro cifre per valori a due byte. Questi valori possono anche essere espressi in decimale, basta delimitare il numero tra due parentesi quadre. I numeri decimali ad un byte possono avere un valore compreso tra —128 e +255; quel-

li a due byte da —32768 a 65535 (—1 corrisponde, in quest'ultimo caso, a 65535). Volendo usare numeri binari od ottali, si può far precedere il numero dal prefisso &B oppure &O, sempre tra parentesi quadre. Nel caso l'istruzione scritta sia in una forma sintattica errata, essa viene automaticamente cancellata, oppure riappare la precedente istruzione, già scritta in quella particolare riga. Se invece l'istruzione è giusta, vengono visualizzati sulla sinistra dello schermo i byte di codifica e si passa alla riga seguente.

Per cambiare una riga già scritta, premere il tasto SELECT; scegliere l'opzione 1 del menu, quindi specificare il numero della linea da correggere. Modificate ora la linea visualizzata aiutandovi con il tasto cursore di sinistra, infine premete ENTER per confermare la correzione, poi SELECT e l'opzione 1 per tornare all'ultima riga del programma (riga vuota), che corrisponde al secondo numero tra parentesi della domanda «linea di partenza?». Finito il programma, premere il tasto SELECT per passare al menu. le scelte possibili sono:

1) Correggere o aggiungere linee in fondo al programma, come già descritto prima.

2) Visualizzare tutto il listato. Il programma chiede il numero di partenza della prima riga da visualizzare; usare il tasto STOP per fermare lo scroll o il tasto SELECT per uscire. In fondo al listato vengono visualizzate le righe ancora libere.

3) Registra il programma come file ASCII in modo da poterlo caricare in seguito in memoria per eventuali modifiche.

4) Recupera il file, senza però effettuare l'assemblaggio.

5) Assembla il programma residente in memoria a partire dall'indirizzo specificato. Se poi il programma dovrà funzionare a partire da un altro indirizzo, lo si potrà definire prima dell'assemblaggio. Finita questa operazione è possibile salvare l'area di memoria contenente il programma eseguibile con il comando BSAVE.

6) Cancella parte del programma. Bisogna specificare il numero della linea dove iniziare a cancellare e quante linee vanno eliminate.

7) Con questa opzione si aggiungono delle righe contenenti dei rem che possono, in seguito, essere modificati. Bisogna specificare il numero delle righe da aggiungere e poi il numero della riga di partenza delle stesse. Non è possibile aggiungere righe lasciandone alcune vuote tra la prima parte del programma e quella aggiunta.

8) Sottomenu label.

9) Registra il programma «ASSEMBLER ESPANSO» insieme alle routine in l/m necessarie al suo funzionamento, nonché quella parte dei 32KRAM inferiori (coperti dalla ROM) che non può essere registrata con le normali istruzioni Basic.

L'opzione @ serve per uscire dal programma e per provare le routine appena assemblate. Si possono definire le routine dal numero 0 al numero 3 (le altre servono al programma assemblatore).

Le opzioni del sottomenu label sono le seguenti:

1) Permette d'inserire delle label assolute, ossia indirizzi che non fanno riferimento a delle istruzioni ma che, ad esempio, servono per delle variabili fisse. I nomi delle label non hanno bisogno del simbolo # come quando vengono usate nella stesura del listato; gli indirizzi assoluti vanno dati in decimale o esadecimale se preceduti dal prefisso &H.

2) Lista le label del punto n.1; alla fine indica quante ne possono ancora essere definite.

3) La funzione di questa opzione è simile alla precedente, con la differenza che ora si tratta delle label incluse nel programma stesso, ossia quelle che fanno riferimento ad istruzioni; quindi il loro indirizzo cambierà a seconda di come il programma verrà assemblato.

4) Cancella una label assoluta della quale conosce il numero.

5) Indica la memoria libera per le stringhe e per le variabili.

Per tornare al menu principale premere il tasto SELECT.

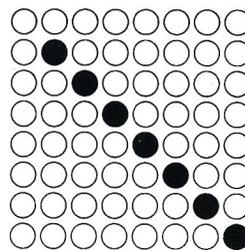
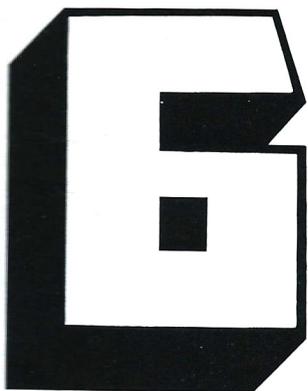
Durante la stesura del programma in assembler, la linea può essere inizializzata con differenti caratteri.

Il carattere ! seguito da quattro cifre esadecimali permette di forzare, da quel punto in poi, un indirizzo di riferimento per le label seguenti, qualunque sia l'indirizzo di assemblaggio scelto. Le label assolute rimangono invariate così come l'indirizzo provvisorio di memorizzazione (partenza reale).

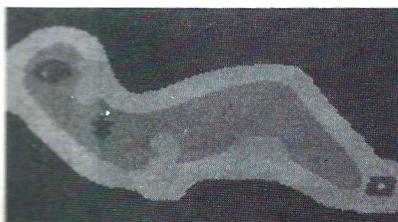
Il carattere @ seguito da una parola lunga da 1 a 4 caratteri memorizza in memoria i codici ASCII dei caratteri scritti. Quando si scrivono questi caratteri, l'inserimento automatico delle maiuscole è disinserito.

Il carattere # seguito da una parola lunga da 1 a 6 caratteri, predispone una label relativa alla prossima istruzione scritta.

Il segno $ corrisponde all'istruzione REM in Basic e permette d'inserire dei commenti.

Il carattere % memorizza il byte esadecimale a due cifre da scrivere di seguito: si possono scrivere fino a 4 byte senza spazi di separazione.

Per usare le label nelle istruzioni assembler occorre farle precedere dal carattere # e racchiuderle tra parentesi quadre; in questo caso, nella codifica a fianco appariranno degli 0 al posto degli indirizzi, o valori che comunque verranno sostituiti dai valori giusti dopo aver eseguito l'assemblaggio del programma. Le label devono avere una lunghezza da 1 a 6 caratteri e, ovviamente, non è possibile scriverne due uguali. Se una label non è stata definita, durante l'assemblaggio appare un apposito messaggio, come nel caso di salti relativi troppo lunghi. Il numero delle label assolute è fissato dalla variabile LB; quello del numero massimo di righe assembler dalla variabile DI. I valori di queste variabili si trovano alla riga 80 del programma Basic.

GOLF 3D

(CLOAD "3DGOLF" - 32 K)

Il golf, attività sportiva oggi molto diffusa, è uno sport che richiede grande preparazione atletica e soprattutto un campo di almeno nove buche. Ecco ora la sorpresa: un programma di simulazione del gioco del golf. Il

programma, oltre che riprodurre le normali situazioni ed anche gli imprevisti di gioco (il vento, le differenti mazze, gli ostacoli. ecc.), offre una piacevole rappresentazione del campo in prospettiva con una buona grafica, propria del sistema MSX. Il gioco si articola lungo una distanza di 4000 metri rappresentati dalle nove buche del percorso.

Alla comparsa del tracciato della prima buca, il programma attenderà l'immissione del tipo di mazza con la quale intendete colpire la pallina. Per la scelta della mazza usate i tasti cursore sinistra/destra e la barra spaziatrice per la selezione. Ricordate che più il numero della mazza è alto, maggiore è l'angolo assunto dalla traiettoria della pallina. Dopo aver selezionato la mazza, sullo schermo si vedrà lampeggiare la lancetta del quadrante inferiore, la quale indica la direzione del colpo. Selezionate la direzione con i due tasti cursore e fissatela con lo spazio. La lancetta del quadrante superiore indica la direzione e la forza del vento: tenetene conto!

L'ultima selezione è quella che rappresenta la forza da imprimere alla pallina, che si ottiene premendo la barra spaziatrice quando il quadratino in movimento ha raggiunto la posizione desiderata (più sarà alta, più forte sarà il colpo).

Attenzione: per caricare il programma procedete in questo modo. Digitate CLOAD "3DGOLF"; Quando appare la scrittta 'OK', fermate il registratore (se non avete inserito la presa REM), digitate RUN e riavviate il registratore per il caricamento del resto del programma.

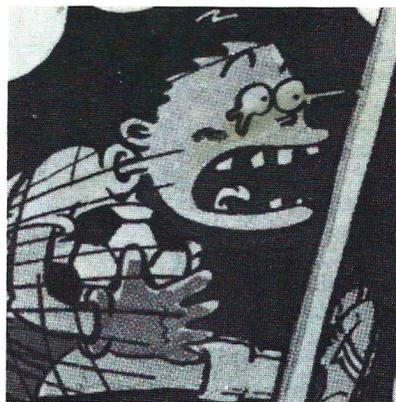
GOAL

(CLOAD "GOAL" - 16 K)

di R. Mesiti

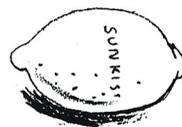
Non si tratta della solita partita di calcio ma di un allenamento specifico in qualità di portiere.

Il gioco consiste nel battere e parare i rigori vicendevolmente se si gioca in due oppure, nell'attesa di trovare un compagno, ci si può allenare a parare avendo come battitore il miglior rigorista del mondo: la CPU.

La sfida a due si svolge secondo le

regole UEFA. Ogni squadra (ogni giocatore, in questo caso) ha a disposizione 5 rigori che batterà uno di seguito all'altro, per meglio adattarsi ai comandi; l'avversario, ovviamente, dovrà cercare di pararli.

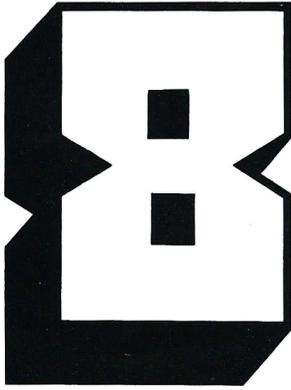
Battuti i 5 rigori, ci si scambiano i ruoli. Se al termine di questa serie il risultato sarà di parità; vi sarà un'altra serie di 3 rigori ciascuno.

Qualora il risultato dovesse finire ancora in parità, si batterà un rigore per parte ad oltranza: cioè fino a quando uno dei due giocatori non sbaglierà determinando la vittoria dell'altro.

Colui che batte potrà governare la potenza e la direzione del tiro. Più si terrà premuta la barra spaziatrice (o

il tasto fire del joystick) tanto più forte sarà il tiro, la cui potenza è visualizzata nell'indicatore in alto a sinistra. Per la direzione si potrà fare uso dei tasti cursore o del joystick: se, per esempio, si sposterà verso sinistra il rigorista, la palla schizzerà dal lato opposto e viceversa.

Bisogna cercare di trovare un buon compromesso tra potenza e direzione perché più il tiro sarà potente, meno sarà preciso (cioè sarà più centrale) e quindi più facilmente parabile.

Chi para potrà scegliere il lato verso il quale tuffarsi usando i tasti cursore (o il joystick) e spiccare il tuffo premendo la barra spazio (o il tasto fire), sempre tenendo premuto il tasto di direzione. Per arrivare agli angoli è necessario spostarsi nella direzione voluta e poi premere la barra spaziatrice (o il tasto fire), senza però rilasciare il tasto direzione.

Buon divertimento!

DOUBLE ROTATION

(CLOAD "DO.RO" - 32 K)

di M. Pessina

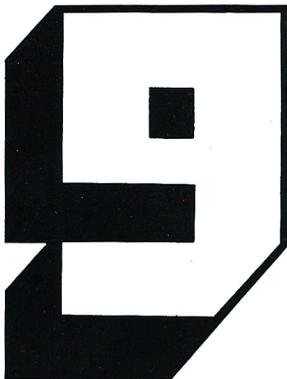
Ecco un gioco, pieno di grafica e con un pizzico di musica, che richiede buone doti di ragionamento. Per caricarlo digitare CLOAD "DO. RO". A caricamento avvenuto (appare la scritta 'OK') digitare RUN ed attendere il caricamento della seconda parte del programma.

Scopo del gioco: ricomporre la schermata grafica divisa, in maniera errata, in dieci parti. Per ricomporre la figura bisogna usare i numeri da 1 a 9. Premendo un numero avremo che: il pezzo corrispondente sarà posto nella prima posizione; il pezzo corrispondente al numero successivo sarà spostato nell'ultima posizione. La posizione degli altri pezzi slitterà verso il numero premuto. Ad esempio, supponiamo di dover ordinare BCDAJEFGHI: premendo 4 avremo che la lettera A verrà spostata nella prima posizione, la lettera J verrà posta nell'ultima posizione e le altre lettere verranno spostate verso la quarta posizione.

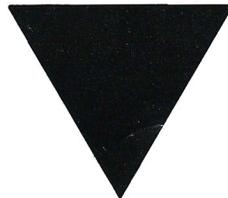
A questo punto avremo le lettere ordinate nella seguente forma corretta: ABCDEFGHIJ.

Per ricomporre la figura avete 999 secondi ed un numero massimo di mosse che dipende dal livello di gioco. Più quest'ultimo salirà, più interessante sarà la figura da comporre...

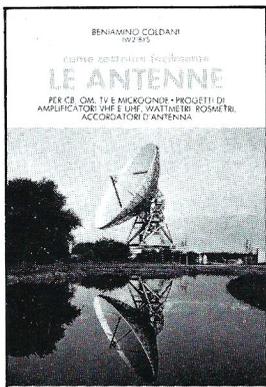
Per giocare porre il tasto CAPS LOCK in on.

■

CALCULATOR

(CLOAD "CALCOL" - 16 K)

di G. Bellomusto

Spesso pretendiamo di attribuire al nostro computer capacità da grande elaboratore elettronico e difficilmente riusciamo a mettere in pratica quello che vorremmo ottenere. Ci dimentichiamo, invece, che l'MSX è un ottimo sistema per i calcoli, o meglio, è un'ottima calcolatrice, soprattutto se si usa questo programma.

Calculator trasforma il computer in una sofisticata calcolatrice. Oltre che le comuni operazioni, è possibile eseguire anche complesse espressioni numeriche basate su funzioni matematiche.

Per usare il programma bisogna scegliere l'opzione, cioè il tipo di calcolo da eseguire, tramite i tasti cursore e convalidare la scelta con la barra spaziatrice. Attenzione, molte operazioni (il coseno, per esempio) richiedono un solo dato, mentre per altre

bisognerà digitare due valori.
Una volta inserito un dato, premere RETURN per confermare l'input. Usando i tasti funzione si opera sulle memorie; dopo l'apparizione del risultato e prima di premere il tasto di spazio, il tutto sarà segnalato da un quadratino presente in basso sullo schermo.

Vedrete: con questo programma risulterà più semplice fare i conti!

di F. Baccani

CORSO DI DATTILO

(CLOAD "CORSO2" - 32 K)

Questa seconda puntata comprende le seguenti lezioni: 6, 7, 8, 9, 10, 21 e 0. Imparerete ad usare ed a memorizzare le locazioni dei caratteri ”5678^*(—TYCVBN>/”.

Con il tasto funzione F5 potrete vedere i vari contatori, utili anche per disputare delle gare di abilità con i vostri amici e per vedere il vostro indice di apprendimento. Con il tasto F3 tornate alla lezione azzerando i vari contatori, mentre con il tasto F1 tornate al menu per le varie opzioni. Se avete il drive, c'è un'altra funzione che potrà tornare utile quando vi trovate al menu e, anziché cambiare lezione, vorrete addirittura passare ad un'altra parte del corso. In tal caso premete F2 e poi return: accederete ad un sottomenu per la gestione dei file su disco, avendo così la possibilità di passare rapidamente da una parte all'altra del corso come se fosse un unico programma. Questo, naturalmente, sarà completamente operativo quando sarete in possesso di tutte e quattro le puntate. Ricordate di salvare le parti del corso con il nome presente nel riquadro, se volete che tutto funzioni nel modo giusto.

Per eseguire correttamente il corso è importante abituarsi a non guardare la tastiera e a tornare con le dita sempre a contatto dei tasti pilota.

BYTE IN LIBERTÀ

Sto cercando di imparare sempre di più a programmare. Vi sarei grato se vorrete cortesemente suggerirmi come utilizzare il programma per l'utilizzo della memoria fantasma e come fare a correggere il programma DUPLI registrato nella cassetta (MSX n. 7).

Domenico Mignosa, Siracusa

Per usare il programma della "memoria fantasma" bisogna caricare il programma in l/m e poi, con le funzioni USR0 e USR1, è possibile scrivere e leggere la memoria fantasma. Per scrivere nell'indirizzo n il dato d bisogna digitare:
A = USRO(n) : A = USR0 (d)
cioè utilizzando sempre la funzione USR0 bisogna prima passare il valore dell'indirizzo e poi il dato.

Per leggere, invece, basta scrivere nella funzione USR1 l'indirizzo di memoria da testare. Volendo quindi

leggere il dato memorizzato nella locazione n scriveremo:
A = USR1 (n)
e nella variabile A avremo il valore contenuto nell'indirizzo n.

Tale indirizzo deve essere compreso tra 0 e 32767.

Per copiare un programma sfruttando l'utility DUPLI bisogna trovare con quest'ultima l'indirizzo d'inizio (i), fine (f) e start (s) del programma da copiare. Dopodiché dare un reset al computer e digitare CLEAR 50, i-1. Poi digitare BLOAD "CAS:" e caricare il programma da copiare. Appena appare la scritta OK salvare il programma su una cassetta vergine con il comando BSAVE"CAS:", i,f,s dove i,f, ed s sono i valori di inizio, fine e start trovati con il programma DUPLI.

GLI INDIRIZZI COME FARE

Avrei un piccolo problema. Quando voglio registrare un programma con il comando BSAVE non so quali indirizzi di partenza dare e nemmeno quelli finali. Potreste aiutarmi?

Andrea Piras - Arbatax

L'istruzione BSAVE serve a registrare su nastro, o su disco, una parte della memoria RAM del tuo MSX. La forma sintattica di tale istruzione è:

BSAVE «nome»,I,F,S,

dove nome è il nome del programma che vuoi salvare, I l'indirizzo d'inizio della memoria da salvare, F quello di fine ed S l'eventuale indirizzo di start di partenza di un programma in linguaggio macchina.

Gli indirizzi sopra elencati hanno un valore dipendente da ciò che vuoi memorizzare. Per esempio: se vuoi salvare una zona di memoria dall'indirizzo 50000 all'indirizzo 51000 scriverai:

BSAVE «pippo», 50000,51000

se invece vuoi salvare un programma in linguaggio macchina (uno dei tanti che abbiamo pubblicato sulla nostra rivista) scriverai:

BSAVE «pippo», I,F,S

dove I, come detto prima, è l'inizio del programma, F l'indirizzo di fine ed S l'eventuale start che può coincidere con I.

IL BASIC PER TUTTI

Mi hanno regalato un MSX e vorrei imparare ad usare il Basic, magari studiandolo con dei corsi completi per MSX. Sapete dirmi se vi sono ed eventualmente gli indirizzi per ottenerli?

Giovanni Finizio — Padova

Purtroppo non vi sono in commercio corsi completi, come tu dici, di Basic per MSX. Noi abbiamo cercato, dal n. 1 al n. 6 della nostra rivista, di dare una spiegazione completa riguardo il linguaggio di programmazione residente nei computer MSX, cioè il Basic MSX. Nel corso abbiamo cercato di mettere più esempi possibile proprio per aiutare il lettore alla sperimentazione. Ti consigliamo di prendere in mano i primi sei numeri di MSX COMPUTER MAGAZINE e di leggere gli articoli sopra descritti. Senz'altro ti saranno di valido aiuto. Per esperienza personale sappiamo

che è anche bene avere a disposizione un buon libro. Noi ci siamo avvalsi de «IL MANUALE MSX» di P.Hoffman edito dalla McGraw-Hill, che potrai trovare in qualsiasi libreria fornita di testi di informatica.

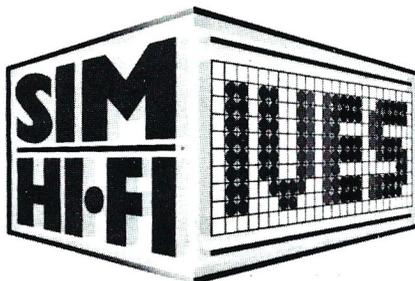
MEMORY CONTROL

Ho da un paio di mesi un computer MSX 2 della Philips Mod VG-8235. Ho acquistato il numero 12 della vostra rivista e purtroppo quando carico il programma «Il gioco dell'oca», sullo schermo compare la scritta «OUT OF MEMORY». Cosa può essere?

Fabrizio Svetina — Trieste

Se ti appare la scritta «OUT OF MEMORY» durante il caricamento de «IL GIOCO DELL'OCA» è possibile, avendo un MSX 2, che non vi sia memoria sufficiente perché in parte usata dal drive. Per superare il problema spegni la macchina, premi il tasto SHIFT o SELECT, oppure CONTROL (prova con ognuno di essi), tienilo premuto (ed accendi il computer) fintanto che non appare la scritta grande «MSX». Ora, se hai a disposizione 28815 byte potrai caricare senza problemi il programma. ■

SIM-HI-FI IVES

SIM HI-FI IVES

3-7 settembre 1987
fiera milano

21° salone internazionale della musica e high fidelity
international video and consumer electronics show

Ingresso:
Porta Meccanica
(Piazza Amendola MM1)
Orario: 9,00 - 18,00

Aperta al pubblico:
3-4-5-6 settembre
Giornata Professionale:
7 settembre
(senza ammissione del pubblico)

Segreteria Generale SIM-HI·FI-IVES - Via Domenichino, 11 - 20149 Milano - Tel. (02) 4815541 - Telex 313627 - Fax (02) 4696055

BIOS UTILITY

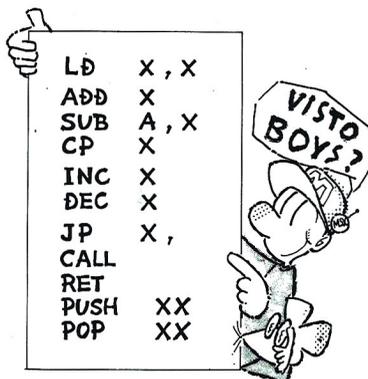
È INUTILE SPRECARE DEL TEMPO PER PROGRAMMARE DELLE ROUTINE IN ASSEMBLER. SONO GIÀ PRONTE ALL'USO NEL BIOS! VEDIAMO COME CHIAMARLE ED UTILIZZARLE

a cura della Redazione

Tanti vorrebbero realizzare delle procedure in linguaggio macchina capaci di gestire parti del sistema, ma spesso la messa in pratica di una cosa siffatta risulta affatto facile.

Per aiutare tutti coloro che si sono trovati ad affrontare problemi di questo genere, e che comunque conoscono le nozioni basilari dell'assembler Z80, spieghiamo come usare alcune parti di programma presenti nella ROM.

IL BIOS

Il Basic Input Operating System (BIOS) è l'insieme dei programmi in linguaggio macchina presenti nella ROM del computer. Questo complesso programma, formato da più routine, gestisce l'intero sistema dell'MSX, cioè il video, la tastiera, la memoria, il suono, etc.

Le routine di lettura della posizione del joystick e del tasto fire, per esempio, appartengono al BIOS. Vi sono tante altre routine di facile utlizzo, basta semplicemente chiamarle puntando al loro indirizzo con l'istruzione assembler CALL ed eventualmente caricando in alcuni registri dei valori opportuni come parametri di input.

Vediamo alcune routine di facile utilizzo e molto efficaci.

IL JOYSTICK

Tutti gli MSX hanno la possibilità di venir collegati ad uno o due joystick. Nella ROM vi sono due routine con le quali è possibile determinare la condizione di ogni joystick. La prima routine restituisce la posizione del joystick n. 1 o n. 2. Per questo, prima

Le frecce indicano le otto posizioni del joystick. Ogni posizione è indicata da un numero compreso tra 1 ed 8. Lo 0 indica che il joystick è fermo.

LIST 1

```
1000 CLEAR 200,39999!
1010 GOSUB 2000
1020 CLS
1030 INPUT "JOYSTICK (1 O 2) ";J
1040 POKE 40001!,J
1050 DEFUSR=40000!
1060 A=USR(0)
1070 B=PEEK(50000!)
1080 PRINT "POSIZIONE JOYSTICK N. ";B
1090 GOTO 1030
2000 RESTORE 2010:FOR N=40000! TO 40008!
:READ A$:POKE N,VAL("&H"+A$):NEXT N:RETU
RN
2010 DATA 3E,01,CD,D5,00,32,50,C3,C9
```

Caricatore Basic della routine che restituisce la posizione del joystick.

di chiamare la routine, bisogna caricare nel registro A il valore 1 o 2. La routine è allocata all'indirizzo 213 e viene chiamata usando l'istruzione assembler CALL 213. A fine esecuzione, nel registro A ritorna un valore compreso tra 0 ed 8 indicante la posizione del joystick all'atto della chiamata della routine. Nella figura 1 è riportato lo schema di ogni posizione che il joystick può assumere e il relativo codice di ritorno nel registro A. Se il valore ritornato è 0 ciò vuol dire che il joystick è in posizione di riposo. Per provare immediatamente la routine non disponendo di un assemblatore, abbiamo pubblicato un programmino in Basic (vedi listato 1) il quale provvede a caricare in memoria, a partire dall'indirizzo 40000, i codici macchina e chiede quale joystick testare, nonché memorizza alla locazione 50000 il valore letto.

La seconda routine relativa al joystick serve per sapere se è stato premuto o meno il tasto fire. La routine è allocata all'indirizzo 216 ed anche qui, prima di essere chiamata, il registro A deve contenere il valore 1 o 2 a seconda del joystick che si vuole testare. Il risultato è posto sempre nel registro A, il quale assume il valore 255 se il tasto è stato premuto; viceversa, il valore contenuto in A sarà 0. Anche in questo caso abbiamo preparato un programmino in Basic (listato 2) che carica da solo i codici macchina. Attenzione: provate il programma solo se avete connesso realmente il joystick al vostro MSX, perché l'unico modo per tornare al Basic è quello di premere il tasto fire.

Vi sono tante altre routine nel BIOS già pronte all'uso. Basta conoscere, come nel caso di quelle che abbiamo appena visto, l'indirizzo di chiamata da passare all'istruzione CALL ed eventualmente i valori di input ed output. Vi proponiamo altre utili procedure. Il numero dopo l'istruzione CALL indica, in decimale, l'indirizzo di partenza della routine.

LIST 2

```
1000 CLEAR 200,39999!
1010 GOSUB 2000
1020 CLS
1030 INPUT "JOYSTICK (1 O 2) ";J
1040 POKE 40001!,J
1050 DEFUSR=40000!
1060 A=USR(0)
1070 PRINT "IL TASTO FIRE E' STATO PREMU
TO"
1080 GOTO 1030
2000 RESTORE 2010:FOR N=40000! TO 40008!
 READ A$:POKE N,VAL("&H"+A$):NEXT N:RETU
RN
2010 DATA 3E,01,CD,D8,00,3C,20,F8,C9
```

Caricatore Basic della routine che testa se è stato premuto il tasto fire del joystick.

CALL 86

La funzione di questa procedura è quella di riempire un blocco della memoria video (VRAM) con un valore costante. Prima di chiamare la routine bisogna predisporre il registro HL contenente l'indirizzo di memoria nel quale iniziare a trasferire i dati. La lunghezza del blocco da riempire deve essere indicata dal registro BC, mentre il registro A deve contenere il valore che deve riempire il blocco di memoria.

CALL 89

Per spostare parte della memoria video (VRAM) nella memoria utente (RAM), (per esempio, per copiare lo screen) bisogna usare questa routine. Prima di chiamarla, il registro HL deve contenere l'indirizzo d'inizio del blocco VRAM da trasferire. Il registro DE, invece, deve indicare l'indirizzo di memoria RAM dal quale iniziare a trasferire i dati; infine, nel registro BC deve essere memorizzato il numero di byte da trasferire.

CALL 92

Esegue l'operazione contraria di quella della CALL 89. Cioè trasferisce un blocco di dati dalla memoria RAM alla memoria video. Il registro BC deve contenere, prima della chiamata della routine, il numero di byte da trasferire. L'indirizzo d'inizio del blocco di memoria in RAM deve essere indicato dal registro HL, mentre il registro DE deve contenere il valore del registro nel quale trasferire i dati in VRAM.

CALL 159

La funzione svolta da questa routine è completamente diversa da quella viste sin'ora. Quando questa routine è chiamata, non ritorna al Basic finché non viene premuto un tasto e restituisce in A il codice ASCII del tasto premuto.

CALL 198

Per posizionare il cursore in una zona del video bisogna usare questa routine. I parametri di input sono: il registro H contenente il numero della colonna nella quale posizionare il cursore e il registro L contenente il numero della riga.

CALL 204

Questa routine non ha nessun registro di input e nemmeno di output perché la sua funzione è quella di cancellare le scritte dei tasti funzione in basso sullo schermo.

CALL 77

Per realizzare la funzione Basic VPOKE (m), n in linguaggio macchina, bisogna usare questa routine. Prima di essere chiamata, il registro HL deve contenere il valore dell'indirizzo della VRAM mentre il registro A deve contenere il valore da scrivere. ■

SOFTWARE

WORD PROCESSOR

ORGANIZZAZIONE GENERALE DI UN PROGRAMMA DI ELABORAZIONE TESTI. LE OPERAZIONI SUL TESTO: DALLA DEFINIZIONE DELLA PAGINA ALLA STAMPA

Terza puntata

di EMANUELE DASSI

Con questo articolo concludiamo la programmazione del Word Processor. L'intera stesura del programma è stata divisa in tre parti:
1. Input da tastiera
2. Output su video
3. Operazioni sul testo

Avendo già parlato, nelle puntate precedenti, dei primi due punti, non rimane che affrontare l'ultimo punto: operazioni sul testo.

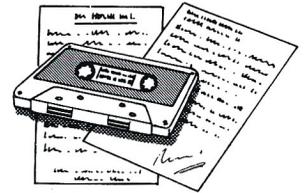
Più precisamente, la parte di programma che commentiamo in questo articolo svolge le seguenti funzioni:
1. Definizione pagine
2. Posizionamento nel testo
3. Ricerca
4. Definizione parametri stampa
5. Stampa
6. Registrazione/caricamento testo
7. Ritorno al Basic

Tutte le funzioni sono state scritte in Basic senza ricorrere a particolari routine in linguaggio macchina. Questa parte di programma viene attivata dall'utente e non ciclicamente dal programma, come avviene per l'input da tastiera e per l'output su video. Per richiamare le operazioni sul testo bisogna premere il tasto SELECT.

RIVEDIAMO L'INPUT DA TASTIERA

Per comprendere il funzionamento di questa parte del WP bisogna rivedere il listato pubblicato nella prima puntata. Alla linea 1070 viene con-

IL PROGRAMMA

```
3000 CLS:INPUT "NUMERO RIGHE PER PAGINA ";NR%
3010 IF NR%<1 THEN 3000
3020 CLS:INPUT "NUMERO COLONNE PER PAGINA ";LP%
3030 IF LP%<20 THEN 3020
3040 RETURN
3050 MP=INT(LT%/(LP%*NR%)+1)
3060 CLS:INPUT "POSIZIONAMENTO ALLA PAGINA ";GP%
3070 IF GP%<1 OR GP%>MP THEN 3060
3080 IL=49999!+(LP%*NR%*GP%):RETURN
3110 CLS:INPUT "INIZIO N' PAGINA ";NP%
3120 IF NP%<1 THEN CLS:GOTO 3110
3130 CLS:INPUT "INTERVALLO TRA LINEE (0-1) ";IL%
3140 IF IL%<0 OR IL%>1 THEN 3130
3150 CLS:INPUT "MARGINE SUPERIORE (0-10) ";MS%
3160 IF MS%<0 OR MS%>10 THEN 3150
3170 CLS:INPUT "MARGINE INFERIORE (0-10) ";MI%
3180 IF MI%<0 OR MI%>10 THEN 3170
3185 RETURN
3190 CLS:INPUT "SAVE O LOAD (S/L) ";R$
3210 IF R$="S" OR R$="s" THEN 3320
3220 CLS:INPUT "NOME DEL TESTO (MAX 6 CARATTERI) ";T$
3230 PRINT "PREPARA IL REGISTRATORE E PREMI UN TASTO"
3235 R$=INPUT$(1):PRINT "PREMI PLAY"
3240 BLOAD "CAS:"+T$
3250 LT%=PEEK(49950!)+256*PEEK(49951!)
3260 NR%=PEEK(49952!)+256*PEEK(49953!)
3270 LP%=PEEK(49954!)+256*PEEK(49955!)
3280 NP%=PEEK(49956!)+256*PEEK(49957!)
3290 IL%=PEEK(49958!)
3300 MS%=PEEK(49959!):MI%=PEEK(49960!)
3310 RETURN
3320 CLS:INPUT "NOME DEL TESTO (MAX 6 CARATERI) ";T$
3325 IF T$="" THEN T$="TESTO"
3330 PRINT "PREPARA IL REGISTRATORE (PLAY+REC) E"
3335 PRINT "PREMI UN TASTO":R$=INPUT$(1)
3340 POKE 49950!,LT%-INT(LT%/256)*256
3350 POKE 49951!,INT(LT%/256)
3360 POKE 49952!,NR%-INT(NR%/256)*256
3370 POKE 49953!,INT(NR%/256)
3380 POKE 49954!,LP%-INT(LP%/256)*256
3390 POKE 49955!,INT(LP%/256)
3400 POKE 49956!,NP%-INT(NP%/256)*256
3410 POKE 49957!,INT(NP%/256)
3420 POKE 49958!,IL%
3430 POKE 49959!,MS%
3440 POKE 49960!,MI%
3450 BSAVE "CAS:"+T$,49950!,49999!+LT%
3460 RETURN
3500 CLS:PRINT "VUOI ABBANDONARE IL PROGRAMMA (S/N)":R$=INPUT$(1)
3510 IF R$="S" OR R$="s" THEN END
3520 RETURN
3530 CLS:INPUT "PAROLA DA CERCARE: ";R$
3540 IF R$="" THEN RETURN
3550 LI=IL
3560 RS$="":IF LI+255<=49999!+LT% THEN TT=LI+255 ELSE TT=49999!+LT%
3570 FOR NN=LI TO TT:RS$=RS$+CHR$(PEEK(NN)):NEXT NN
3580 PS%=INSTR(RS$,R$)
3590 IF PS%<>0 THEN IL=LI+PS%-1:RETURN
```

Listato 1.
Il programma relativo alle operazioni sul testo è formato da diverse subroutine.

frontata la variabile CRK% con il valore 24 (codice ASCII del tasto SELECT) per vedere se è stato premuto il tasto SELECT. In caso affermativo, viene chiamata la subroutine alla linea 4000 la quale attiva i vari comandi sul testo.

Il listato n.2 riproduce la subroutine in questione, difatti inizia con il numero di linea 4000. Il suo funzionamento è semplice. Questa subroutine viene chiamata quando si preme il tasto SELECT. Presenta un menu con

```
3600 LI=TT+1:IF LI>49999!+LT% THEN RETURN ELSE 3560
3610 CLS:PRINT "ACCENDI LA STAMPANTE E PREMI UN TASTO":R$=INPUT$(1)
3620 INPUT "PAUSA A FINE PAGINA (S/N) ";R$:LI=50000!:PN%=NP%
3630 IF R$="S" OR R$="s" THEN P$="S" ELSE P$="N"
3640 FOR NN=1 TO MS%
3650 LPRINT
3660 NEXT NN
3670 FOR NN=1 TO NR%
3675 IF LI>49999!+LT% THEN R$=STRING$(LP%," "):GOTO 3700
3680 IF LI+LP%-1>49999!+LT% THEN GOSUB 3800:GOTO 3700
3690 GOSUB 3850
3700 LPRINT R$:IF IL%=1 THEN LPRINT
3710 NEXT NN
3720 FOR NN=1 TO MI%
3730 LPRINT
3740 NEXT NN
3750 LPRINT STRING$(INT(LP%/2)," ");PN%:PN%=PN%+1
3760 IF LI>49999!+LT% THEN RETURN
3770 IF P$="S" THEN PRINT "CAMBIA FOGLIO E PREMI UN TASTO":R$=INPUT$(1):CLS
3780 GOTO 3640
3800 R$=""
3810 FOR N1=LI TO 49999!+LT%
3820 R$=R$+CHR$(PEEK(N1))
3830 NEXT N1
3840 LI=50000!+LT%:RETURN
3850 R$=""
3860 FOR N1=LI TO LI+LP%-1
3870 R$=R$+CHR$(PEEK(N1))
3880 IF PEEK(N1)=13 THEN LI=N1+1:RETURN
3890 NEXT N1
3900 FOR N1=LI+LP% TO LI STEP -1
3910 IF MID$(R$,N1-LI,1)=" " THEN R$=LEFT$(R$,N1-LI):LI=N1:RETURN
3920 NEXT N1:RETURN
```

tutte le opzioni relative alle operazioni sul testo ed esegue la subroutine relativa alla scelta dell'utente (linee 4070 e 4080). In sintesi, la parte di programma dalla linea 4000 alla linea 4090 non fa nient'altro che richiamare un'altra subroutine in base all'operazione che si vuole fare sul testo.

IL LISTATO

Il listato n. 1 contiene tutte le istruzioni per sviluppare le sette operazioni sul testo. È piuttosto corto perché siamo riusciti ad ottimizzare l'uso delle variabili e delle istruzioni. Il listato inizia alla linea 3000 e termina con la linea 3920. Non è stato rispettato l'incremento costante tra i numeri di linea perché, non trattandosi di un'unica subroutine, abbiamo lasciato più spazio tra quelle che potrebbero essere espanse in futuro.

Vediamo di commentare il listato considerando le parti di programma relative ad ogni funzione.

DEFINIZIONE PAGINA

Nella scorsa puntata abbiamo presentato, oltre che la subroutine della gestione del video, anche le linee 10-60 le quali chiamano le varie parti del WP ed inizializzano alcune variabili (linea 40) tra le quali LP% e NR% che indicano, rispettivamente, la larghezza della pagina espressa in numero di colonne ed il numero di righe in una pagina.

Dalla linea 3000 alla linea 3040 vi è

```
4000 CLS:PRINT "1- DEFINIZIONE PAGINA"
4010 PRINT "2- POSIZIONAMENTO NEL TESTO"
4020 PRINT "3- RICERCA"
4030 PRINT "4- DEFINIZIONE PARAMETRI STAMPA"
4040 PRINT "5- STAMPA"
4050 PRINT "6- SAVE/LOAD"
4060 PRINT "7- RITORNO AL BASIC"
4070 PRINT "SCEGLI L'OPZIONE":R$=INPUT$(1)
4080 ON VAL (R$) GOSUB 3000,3050,3530,3110,3610,3190,3500
4090 RETURN
```

Listato 2.
Questa subroutine è chiamata quando viene premuto il tasto SELECT e, a sua volta, richiama la subroutine delle operazioni sul testo.

la subroutine che definisce le dimensioni della pagina del testo. La sua funzione è quella di poter cambiare i valori delle variabili LP% e NR%.

POSIZIONAMENTO NEL TESTO

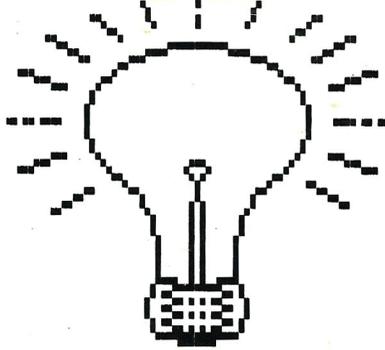
Quando si scrive un testo piuttosto lungo o che comunque non può essere visualizzato completamente sul video, può essere utile poter rivedere una parte del testo specificando direttamente la pagina da visualizzare. Le linee 3050-3080 svolgono questa funzione. La variabile GP% indica la pagina da visualizzare, mentre MP fornisce il numero massimo di pagine scritte. La visualizzazione della nuova pagina è determinata dalla modifica del valore della variabile IL; quest'ultima è stata descritta insieme alla subroutine per la gestione del video.

RICERCA

Nonostante si siano usate poche linee di programma (dalla linea 3530 alla linea 3600), siamo riusciti a realizzare una tra le funzioni più importanti in un WP: la ricerca. Capita, infatti, di voler localizzare nel testo una particolare parola, un numero o addirittura una frase, senza necessariamente leggere tutto quanto è stato scritto. La ricerca ha luogo a partire dal primo carattere visualizzato in alto a sinistra sul video e si ferma non appena trova quanto cercato. Per proseguire nella ricerca bisogna nuovamente attivare questa funzione senza aver fatto scorrere il testo sul video. Questo fatto va tenuto in considerazione perché la logica di ricerca si basa sulla variabile IL che, come già detto, indica il carattere in alto a sinistra del video. La ricerca della stringa si avvale della funzione Basic INSTR, la quale opera su lunghezze di 255 caratteri dell'intero testo.

Consigliamo il lettore di soffermarsi su questa parte di programma perché è un ottimo esempio di sintesi di programmazione.

DEFINIZIONE PARAMETRI DI STAMPA

Abbiamo voluto inserire alcuni parametri di stampa che non sempre sono presenti in tutti i WP: numero iniziale da stampare sulle pagine, margine superiore ed inferiore, ecc... Poiché alla fine il risultato di un programma di questo genere è la stampa del testo inserito, riteniamo importante dare prestazioni di questo genere.

Le linee 3110-3185 servono a definire i parametri di stampa. Alla domanda «Inizio n' pagina?» si deve intendere il numero dal quale iniziare a numerare le pagine stampate. Può essere utile definire un numero superiore ad 1 se quanto verrà stampato è la continuazione di un testo elaborato in precedenza. Successivamente verrà chiesto l'intervallo tra la stampa di una linea e un'altra. Tale intervallo può esserci o meno. Se c'è, corrisponde ad una linea bianca. Infine, vengono chiesti i margini superiore e inferiore, cioè il numero delle linee bianche da lasciare in cima e in fondo ad ogni pagina. Si può inserire un valore compreso tra 0 e 10. Rispondendo con 0 non verrà lasciato nessun margine. A differenza di altri programmi di WP, quando si definisce il numero di righe per pagina (per intenderci la variabile NR%) si deve considerare solo il numero di righe di testo, senza eventuali intervalli e margini. Le variabili utilizzate sono quattro: NP% indica la numerazione della pagina, IL% l'intervallo tra la stampa di una linea ed un'altra, MS% il margine superiore ed MI% il margine inferiore.

STAMPA

Dalla linea 3610 alla linea 3920 si articola la parte più complessa del programma delle operazioni sul testo: la stampa. Oltre che eseguire la stampa del testo presente in memoria, considerando i vari parametri definiti, è anche possibile ordinare la pausa tra la stampa di una pagina e quella successiva. Questo si rivela utile quando si utilizzano fogli singoli. La sobroutine non stampa su una riga esattamente tanti caratteri quanti sono quelli definiti dalla variabile LP%, ma cerca di avvicinarsi il più possibile a tale valore senza superarlo. Per calcolare i caratteri da stampare su una linea, utilizza altre due subroutine (gosub 3800 e gosub 3850). La linea da stampare è calcolata sulla base degli spazi compresi tra le parole e viene memorizzata nella variabile R$. Successivamente viene inviata alla stampante con l'istruzione Basic LPRINT R$.

REGISTRAZIONE E CARICAMENTO TESTO

Naturalmente non poteva mancare l'opzione di save/load testo, indispensabile per lavorare in tempi differenti. Alla linea 3190 inizia la parte di programma comune sia al save che al load poi, fino alla linea 3310, si sviluppa la procedura di caricamento, mentre le linee 3320-3460 si occupano della registrazione del testo. L'operazione di save chiede di assegnare un nome al testo da salvare. Se si risponde con una stringa nulla, automaticamente il programma provvede ad assegnare al file il nome «TESTO». Nel file vengono registrate anche alcune variabili, in particolare LT% che indica la lunghezza del testo. NR%, LP%, NP%, IL%, MS%, MI%; queste ultime definiscono il formato della pagina da stampare. Naturalmente, in fase di caricamento tali variabili vengono ripristinate (linee 3250-3300).

RITORNO AL BASIC

Riteniamo che non sia il caso di spiegare questa subroutine (linee 3500-3520) la cui funzione è piuttosto ovvia. Ci sembra invece doveroso concludere questo articolo facendo alcune precisazioni.

Il programma WP che abbiamo presentato in queste tre puntate ha avuto lo scopo di dimostrare che anche i programmi complessi di questo genere possono essere realizzati da soli, scomponendo in più problemi la funzione dell'intero programma. Certo bisogna sempre tenere presente i limiti di progettazione; nel nostro caso, la lentezza dovuta al Basic come linguaggio di programmazione. Numerose possono essere le migliorie da apportare a questo programma. In primo luogo, l'installazione della possibilità di lavorare in modo «insert», cioè l'inserimento tra due caratteri di una parola. Poi, il miglioramento della centratura del testo durante la fase di stampa. Infine... lasciamo pensare a voi lettori... cosa ancora modificare!

■